AMENDMENTS TO THE CLAIMS:

This listing of claims replaces all prior versions and listings of claims in the application.

LISTING OF CLAIMS:

1. (Original) A method of determining whether a plurality of datapaths executing a computer program should execute conditional processing in the computer program, comprising:

determining whether PE states of all of the datapaths are disabled;

determining whether the computer program is deterministic; and

branching around the conditional processing if the PE states of all of the datapaths are disabled and the computer program is non-deterministic.

2. (Original) The method of claim 1 wherein determining whether the PE states of all of the plurality of datapaths are disabled comprises:

evaluating a processor enable bit associated with each one of the plurality of datapaths.

3. (Original) The method of claim 2 wherein the processor enable bit is enabled if it is a value of one.

4. (Original) The method of claim 2 wherein the processor enable bit is disabled if it is a value of zero.

5. (Original) The method of claim 1 wherein the determining of whether the computer program is deterministic comprises evaluating a deterministic bit.

A2

6. (Original) The method of claim 5 wherein the deterministic bit contains a first value indicating the computer program is deterministic.

7. (Original) The method of claim 5 wherein the deterministic bit contains a second value indicating the computer program is non-deterministic.

8. (Currently Amended) A method of determining whether ~~a plurality of~~ datapaths ~~executing~~ in a program should execute a conditional processing block in the program, the method comprising:

storing states of the datapaths;

determining whether all ~~PE states of~~ the datapaths are disabled; ~~and~~

branching around the conditional processing block if ~~the PE states of~~ all the datapaths are disabled;

executing the conditional processing block if all the datapaths are not disabled; and

setting the states of the datapaths to stored states of the datapaths following executing the conditional processing block.

9. (Original) The method of claim 8 wherein determining further comprises:

determining whether the program is non-deterministic.

A2

10. (Currently Amended) The method of claim 9 wherein ~~the~~ branching further comprises:

branching if the ~~PE~~ states of all of the datapaths are disabled and the program is non-deterministic.

11. (Currently Amended) The method of claim 9 further comprising not branching if the program is deterministic;

wherein the conditional processing block is executed regardless of whether all the datapaths are disabled.

12. (Currently Amended) An instruction set that is stored in a machine-readable medium and that is executed by datapaths during conditional processing, the instruction set comprising one or more instructions ~~an instruction that causes the datapaths~~ to:

store states of the datapaths;

determine whether ~~PE states of~~ all of the datapaths are disabled, ~~and~~

branch around the conditional processing if ~~the PE states of~~ all of the datapaths are disabled;

execute the conditional processing block if all the datapaths are not disabled; and

set the states of the datapaths to stored states of the datapaths following executing the conditional processing block.

13. (Currently Amended) The instruction set of claim 12 wherein the branching is not performed if the program is deterministic; and

wherein the conditional processing block is executed regardless of whether all the datapaths are disabled

14. (Currently Amended) An instruction set stored on a machine-readable medium that is executed by datapaths during conditional processing, the instruction set comprising one or more instructions ~~an instruction that causes the datapaths~~ to:

~~establish a state of~~ store PE states of the datapaths prior to ~~for the~~ conditional processing[[,]];

determine whether ~~the established PE states of~~ all of the datapaths are disabled based on PE states of the datapaths[[,]]; ~~and~~

branch around the conditional processing if ~~the established PE states of~~ all of the datapaths are disabled; and

set the states of the datapaths to stored states of the datapaths if the conditional processing block is executed.

15. (Original) The instruction set of claim 14 wherein the conditional processing includes an if-processing block.

16. (Cancelled)

A2

17. (Original) The instruction set of claim 14 wherein the conditional processing includes an else-processing block.

18. (Original) The instruction set of claim 14 wherein the branching is not performed if the program is deterministic.

19. (New) The method of claim 8, wherein storing and branching are executed via a single instruction.

20. (New) The instruction set of claim 12, wherein storing and branching are executed via a single instruction.